METHOD AND SYSTEM FOR CREATING A GRAPHIC USER INTERFACE FOR CONTROLLING, CONFIGURING AND/OR DIAGNOSING AN ELECTROPHOTOGRAPHIC COPYING OR PRINTING APPARATUS

5

The invention concerns a method and a system for generation of a graphical user interface for an electrophotographic printing or copying system. Data for generation of a graphical user interface are stored in a storage region of a first data processing unit.

10

In known printers or copiers, a serial interface is provided for connection of a service and maintenance computer with which diagnostic tasks and adjustment tasks can be implemented. Individual components of the printer or copier have their own data interfaces at which the service and maintenance computer for

15   diagnostic, configuration and maintenance tasks can be connected via an interface converter. In known printers, the diagnosis and adjustment of these components can only occur via the respective data interface of the component. Due to the relatively low data transfer rates of these data interfaces, larger quantities of data would have to be read out from the printer or copier with the aid of exchangeable

20   data media such as, for example, diskettes. Different programs that are respectively individually invoked and executed by the service and maintenance computer are necessary for evaluation of these error data, for diagnosing and maintenance of the components with their own data interfaces and for diagnosis of the printer or copier via the serial interface. The necessary programs are installed

25   on the service and maintenance computer, whereby different programs that are respectively stored as a separate program on a fixed disc of the service and maintenance computer are necessary for individual delivery states of individual components and of the printer or copier. Even the selection of the correct program requires significant expertise.

30

A control system for a printer or copier is known from the US patent 5,243,382, in which a portable maintenance device can be connected to a maintenance interface. The printer or copier transfers first data with state information of the printer or copier with the aid of a connection between maintenance device and the printer or

5 copier. Furthermore, second data that contain state information can be input to the maintenance device. At least one pair of items of stored control information can be output by the maintenance device based on the first and second state information. Furthermore, from the US patent 5,243,382 it is known to transfer the stored data to a data processing system.

10

Furthermore, from the document EP 0 843 230 A1 a system is known for remote maintenance via the World Wide Web of a peripheral device connected with a network. In this system, a Java applet is downloaded given the first invocation call by a server to a data processing system, and there it is cached by a browser

15 program module for a repeated invocation by the same HTML document. A graphical user interface is generated with the aid of the HTML document, whereby the HTML document is also transferred to the data processing system. A method and an arrangement for implementation of monitoring and management functions in networks with monitored components is known from the document DE 197 35

20 947 A1 and a data transfer over the Internet with the aid of program modules based on the programming language Java is known from the document US 5,926,631 A1.

It is the object of the invention to specify a method and a system for generation of a graphical user interface for an electrophotographic printing or copying system via

25 which a user interface adapted to the respective printing or copying system can be simply generated and at least one operating function and/or diagnosis function is provided for operation or, respectively, for diagnosis of the printing or copying system.

SUBSTITUTE PAGES

The object is achieved via a method with the features of the patent claim 1. Advantageous developments of the invention are specified in the dependent patent claims.

5   A first aspect of the invention concerns a method for loading of program data for operation and/or for diagnosis of an electrophotographic printer or copier. First

data are stored in a first storage region of a first data processing unit of a printer or copier. The first data are transferred from the first data processing unit to a second data processing unit of an operating unit. The first data contain at least specifications about at least one program module necessary for generation of

5 operating and/or diagnosis functions. With the help of the second data processing unit, it is checked whether second data that contain the program module are contained in a second storage region of the second data processing unit. Given nonexistent second data in the second storage region, the second data are transferred from a third storage region of the first data processing unit to the

10 second data processing unit. Instructions of the program module are executed by the second data processing unit.

Via this inventive method for loading of program data, it is achieved that at least the second data can be stored in an operating unit, whereby with the aid of the first

15 data it can simply be checked whether the necessary program module is contained in the second storage region. Only when the necessary program module is not contained in the second storage region are the second data that contain the program module transferred from the first data processing unit to the second data processing unit. The necessary transfer time, in particular for loading of the second data, is

20 thereby significantly shortened, primarily when data lines with a low data transfer rate are used for data transfer.

A second aspect of the invention concerns a system for loading of program data for operation and/or for diagnosis of an electrophotographic printer or copier. The

25 system contains a first data processing unit of the printer that is connected with a second data processing unit of an operating unit via a data line. First data that at least contain specifications about at least one program module necessary for generation of operating and/or diagnosis functions can be transferred from the first data processing unit to the second data processing unit. The second data

30 processing unit checks whether second data that contain the program module are stored in a second storage region of the second data processing unit. In nonexistent

second data in the second storage region, the second data are transferred from a third storage region of the first data processing unit to the second data processing unit. The second data processing unit executes instructions of the program module. It is thereby achieved that the second data do not have to be transferred

5   from the printer or copier when they are already stored in a storage region of the operating unit. Wait times for loading of the data are reduced.

A third aspect of the invention concerns a method for generation of a graphical user interface for an electrophotographic printing or copying system. First data for

10   generation of a graphical user interface are stored in a storage region of a first data processing unit of the printing or copying system. The first data are transferred to a second data processing unit of an operating unit that is connected with the first data processing unit via a data line. The first data are processed by the second data processing unit. The second data processing unit executes a display program

15   module that processes the first data. Second data that are transferred to the second data processing unit are stored in a second storage region of the first data processing unit. The second data are processed by the second data processing unit, whereby at least one operating function and/or diagnosis function is provided for operation or, respectively, for diagnosis of the printing or copying system.

20

Via the method according to the third aspect of the invention, it is achieved that a graphical user interface for operation and/or for diagnosis of the printing or copying system is generated on an operating unit without an installation of special software and an adaptation of program modules being necessary. A printing or

25   copying system-specific adaptation of the operating unit can thus be foregone. A selection from a plurality of programs of a program adapted for the printing or copying system is also not necessary.

The data necessary for generation of the graphical user interface are stored in the

30   printing or copying system itself. These data are already adapted to the type and the output state of the printing or copying system or, respectively, of components

of the printing or copying system, whereby a correct operation, correct configuration and/or a correct diagnosis of the printing or copying system is possible in a very simple manner.

5   A fourth aspect of the invention concerns a system for generation of a graphical user interface for an electrophotographic printing or copying system. A first data processing unit of the printing or copying system contains a first storage region in which are stored first data for generation of a graphical user interface. The first data are transferred to a second data processing unit of an operating unit, whereby

10   the second data processing unit is connected with the first data processing unit via a data line. The second data processing unit processes the first data with the aid of a display module that generates a graphical user interface. Second data that contain program elements are stored in a second storage region of the first data processing unit. The second data are transferred to the second data processing unit

15   via the data line. The second data processing unit processes the second data and provides at least one operating and/or diagnosis function for operation or, respectively, for diaphragm of the printing or copying system.

Via this inventive system it is achieved that the operating unit requires not program

20   elements and program modules specially adapted to the printing or copying system in order to output a user interface or, respectively, user interfaces adapted for the respective printing or copying system. Both the data for generation of a graphical user interface and program elements for operation and/or for diagnosis of the printing or copying system are stored in a storage region of the printing or copying

25   system, whereby both the graphical user interface and the program elements are tuned to the type and/or the output state of the printing or copying system.

For better understanding of the present invention, reference is made in the following to the preferred exemplary embodiments shown in the drawings, which

30   preferred exemplary embodiments are described using specific terminology. However, it is to be noted that the protective scope of the invention should not

thereby be limited, since such changes and further modifications to the shown devices and/or the methods as well as such further applications of the invention as they are presented therein are viewed as typical present or future expert knowledge of a competent average man skilled in the art. Figures show exemplary

5   embodiments of the invention, namely:

Figure 1   a block diagram of a system with a printer and a service and maintenance computer according to a first exemplary embodiment of the invention;

10

Figure 2   a block diagram with a detail view of an input and output controller of the printer and of the service and maintenance computer according to Figure 1;

15   Figure 3   a block diagram that shows the data transfer and communication structure between the service and maintenance computer and the printer according to Figure 1;

Figure 4   an arrangement of a printer and a service and maintenance computer

20   according to a second exemplary embodiment of the invention;

Figure 5   an arrangement of the printer and of the service and maintenance computer according to Figure 4, whereby the service and maintenance computer is connected with the printer via a remote

25   data transfer connection;

Figure 6   a block diagram in which the access of a display program module to program elements is shown;

| Figure 7 | a block diagram with a data processing unit of a service and maintenance computer and a printer controller, whereby the data transfer for transfer of display data is shown; |

5 Figure 8   a service and maintenance computer that is connected with a first and a fifth printer;

Figure 9   a service and maintenance computer that is connected with a second printer; and

10

Figure 10   program data archive for generation of a graphical user interface for configuration and diagnosis.

A block diagram of a system 10 for maintenance, configuration and diagnosis of a

15   printer 12 with the aid of a service and maintenance computer 18 is shown in Figure 1. The printer 12 contains a printer controller 14 and an input and output controller 16. The device controller 14 contains all information and measurement values necessary for control, maintenance, configuration and error analysis as well as for diagnosis of the printer 12. The device controller 14 is connected with the

20   input and output controller 16. A first service and maintenance computer 18 can be connected with the printer 12 via the first network connection 16. The service and maintenance computer 18 is, for example, a service notebook that a service technician connects with the input and output controller 16 of the printer 12 given service assignments on site at the printer 12, for example at a customer. The

25   connection preferably occurs with the aid of a network connection to a local area network (LAN). For this, the service technician connects the service notebook 18 with a first network connection 20 of the input and output controller 16 with the aid of a network cable, what is known as a patch cable. Furthermore, the input and output controller 16 comprises a second network connection 22 that is connected

30   with a network 24 through the printer 12. An external control panel 26 and a LAN modem 28 are connected with the printer 12 via the network 24. The LAN modem

28 is connected with a telephone network 30, whereby a data connection with a service center of the printer manufacturer and with a telephone connection of a service technician can be established via the LAN modem 28. The telephone connection of the service technician and the service center, which is respectively

5   connected with the telephone network 30, is not shown in Figure 1.

Operating personnel can implement operator control actions with the aid of the external control panel. The operating personnel can thus, for example, input paper parameters, position print images, establish print qualities and coordinate the

10   execution of print jobs. With the aid of the service notebook 18, both the operator control actions that operating personnel can execute via the external control panel 26 can be implemented and setting values and parameters of the printer 12 can be changed and storage regions and registers of the device controller 14 can be output and written with new data. In the same manner as with the help of the service

15   notebook 18, respective functions for diagnosis and for adjustment of print parameters are available to an service technician given a connection of a further service notebook to the telephone connection and to a service and maintenance computer in a service center of the printer manufacturer that is connected with the telephone network via in [sic] modem. Thus a remote maintenance, remote

20   configuration and remote diagnosis of the printer 12 is possible via the telephone network 30 and a local diagnosis and maintenance is possible with the aid of the service notebook 18 connected with the printer 12.

A section of the block diagram according to Figure 1 is shown in Figure 2 with the

25   service notebook 18 and the input and output controller 16. Identical elements have the same reference characters. In addition to an operating system, the service notebook 18 contains a display program module 32 that is also designated as a browser program module. The browser program module 32 contains what is known as a Java runtime program environment via which it is possible to executed

30   program elements that have been created in the programming language Java,

whereby the data generated by these program elements are output with the aid of the browser program module 32.

The input and output controller 16 contains what is known as an HTTP server 36 that contains at least one data storage region 38 in which are stored hypertexts and elements for generation of what are known as Hypertext Markup Language pages (HTML pages), whereby the stored data preferably contain what are known as hypertexts for page description. A graphical user interface in which a plurality of user interfaces can be selected, preferably via a menu, is generated with the aid of these HTML pages. The user interfaces are output in a section provided in the graphical user interface.

Furthermore, the HTTP server 36 contains a storage region 40 in which program elements are stored that have been created in the programming language Java. Such a program element is also designated as a Java applet. The program data of the Java applet as well as class data that are necessary for execution of the Java applet and program data of program parts that are invoked by the Java applet are preferably stored in archives, what are known as jar archives. Data n [sic] that are necessary for generation of a function or of a group of functions are preferably stored in an archive. In other embodiments, storage region 40 also stores Java applications as well as program elements that are designated as ActiveX program elements. A plurality of archives with Java applets or, respectively, ActiveX elements are preferably contained in the storage region 40. An archive with program data for authentication, an archive with program data for communication control between the service notebook 18 and the input and output controller 16 as well as further archives with program data for provision of operation, configuration and diagnosis functions is [sic] preferably contained.

The transfer of data with setting values and printer parameters between printer 12 and service notebook 18 preferably occurs with the aid of a Remote Method Invocation (RMI) communication. With the help of such an RMI communication,

it is also possible to access objects of the device controller 14 and a databank (not shown) of the printer 12, preferably a management information base (MIB) of the printer 12. With the aid of the Remote Method Invocation (RMI) communication technique, data can be transferred between objects defined in the programming

5   language Java. The Remote Method Invocation (RMI) communication technique is described in detail in the documents US 6,571,274 B1 as well as US 6,334,146 B1, whose contents are herewith incorporated by reference into the present specification. If SNMP instructions are transferred over a wide area network (WAN) as data with the aid of the RMI communication technique, the

10   manipulation of these SNMP instructions is possible via the relatively secure data transfer with the aid of the RMI communication, since (with the aid of the RMI communication) data can be transferred directly between the objects defined in the programming language Java. The SNMP instructions are thus transferred as useable data and cannot be viewed by third, unauthorized users.

15

Program data or, respectively, archives with program data for diagnosis of a paper input unit, for diagnosis of a paper path controller, for diagnosis of a paper output unit or for diagnosis of a printing unit can be stored in the storage region 40. Furthermore, archives with program data for access to an event registration and

20   archives with program data for access to an error storage are contained in the storage region 40.

A connection is established between the service notebook 18 and the network connector 20 via the network connection with the aid of a Transmission Control

25   Protocol/Internet Protocol (TCP/IP). An HTML page is requested from the HTTP server via a preset in the browser program module 32. The data of the HTML page are read out from the storage region 38 and transferred to the browser program module 32. A Java applet is requested at the HTTP server 36 via execution of the transferred hypertext of the HTML page. The HTTP server 36 reads data from the

30   storage region 40 that contain program data of the Java applet. The program data

of the Java applet are transferred to the service notebook 18 and executed by the Java runtime program environment 34.

In the simplest case, the HTML page or, respectively, the transferred hypertext
5   contains only the instruction for downloading of one or more Java applets. The graphical user interface, display elements and operating elements as well as the operating, configuration and diagnosis function are then realized (as already mentioned) with the aid the java applet program elements.

10   In other exemplary embodiments, given the establishment of the TCP/IP connection it is checked whether the service technician is authorized to receive access to data of the HTTP server 36 with the service notebook 18. The Java applet generates the graphical user interface with the aid of the browser program module 32 and establishes a logical connection with an RMI server 36 with the aid
15   of an RMI communication. The service notebook 18 is an RMI client in the RMI communication. The external operating unit 26 and the service computer connected over the telephone network 30 are also RMI clients. A file transfer protocol (FTP) for data transfer is used for transfer of mass data such as, for example, error data and what are known as trace data. For this, a special FTP
20   server is provided in the input and output controller 16, which FTP server is, however, only activated for data transfer and is otherwise not active for data protection reasons.

To activate the FTP server, with the help of the RMI communication data are
25   transferred via which the input and output controller 16 generates what is known as an SNMP command (SNMP = Simple Network Management Protocol). SNMP commands that are generated with the aid of the RMI communication between the service notebook 18 and the printer 12 are likewise used for transfer of instructions to control units of the printer 12, the readout of parameters from the printer 1, such
30   as sensor values, error states and status information. Variables, parameters, command registers and status registers contained in the already-mentioned

management information base (MIB) can also be accessed with the aid of SNMP commands, whereby an object identifier (OID) is associated as an addressing address with each element of the management information base. The SNMP commands are generated in the RMI server with the help of the transferred RMI

5   commands and RMI data. Thus no direct accesses are possible with external SNMP commands to the MIB of the printer 12. It is thereby ensured that unauthorized people and control units have no access to the controllers 16 and the database of the printer 12. However, the data are directly transferred in the transfer of the data with the aid of the FTP protocol, whereby special measures are then

10   taken in order to prevent an unauthorized access.

A block diagram is shown in Figure 3 in which is shown the RMI/SNMP communication between the printer 12 and the service notebook 18. As already explained in connection with Figure 2, the communication between the service

15   notebook 18 and the printer 12 occurs via a network connection with the aid of an RMI communication. The display program 32 of the service notebook 18 operates a Java applet 48 for generation of an RMI client of an RMI communication. The Java applet 48 determines the SNMP data from the data transferred with the aid of the RMI communication. The SNMP data in turn contain data with which an OID

20   address is respectively associated, for example variable values, measurement values and setting values of the printer 12. The data of individual variables are addressed in a management information base (MIB) with the aid of the OID address. A variable name is then associated with these data by a program function 46, dependent on the OID address. The transferred data are requested and

25   processed by a program function 44 with the aid of the variable names.

The program function 44 serves as an SNMP manager for execution and for generation of SNMP commands as well as for caching of data. The data determines with the aid of the program function 44 and SNMP commands are

30   converted into display data (that are output on a display unit of a service notebook

18) with the aid of a program function 42 for generation of a graphical user interface.

Furthermore, operator control actions and inputs of setting values that are then

5   converted into SNMP commands with the aid of the program function 44 can occur via the graphical user interface. Variable addresses, i.e. OID addresses, are correspondingly associated by the program function 46 with the variable names contained in these commands, whereby the data for generation of the SNMP commands are transferred to the printer 12 together with the variable addresses

10   with the aid of the RMI communication. The data for generation of the SNMP commands in particular contain RMI commands. The input and output controller 16 of the printer 12 contains a program module for generation of SNMP commands from the data transferred with the aid of the RMI communication. The SNMP commands thereby also contain data with setting values. The SNMP commands in

15   particular comprise a get command for retrieval of data from the management information base, a set command for modification of data in the management information base and a trap command for direct transfer of information to a SNMP agent or a SNMP subagent. The SNMP agents are respectively contained as program elements in control units of the printer 12 such as, for example, the device

20   controller 14. The program data of a program function 42, 44, 46 are respectively contained in an archive and can thus be individual transferred, executed and stored.

The input and output control unit 16 transfers the SNMP commands to a SNMP master agent 52 that, with the help of the address associated with the respective

25   command, forwards them to the appertaining control unit in which an element with this address is stored. This forwarding of the command to the respective control unit, of which the control unit 54 and the control unit 14 are shown in Figure 3, is also designated as routing.

30   The SNMP agents 52, 54, 14 as well as the variables and data administered with the aid of the SNMP agents are preferably hierarchically organized such that an

address structure according to the structure of the corresponding SNMP agents is associated with them, whereby the storage location of the respective variables can be simply simply [sic] found with the aid of the OID address. Both diagnosis functions of the individual control units 14, 16, 52, 64 can be invoked with the aid

5 of the SNMP commands and operator control actions can be executed. The data transfer between the graphical user interface 42 and the control units 14, 16, 52, 54 of the printer 12 thus occurs with the aid of SNMP commands that are transferred or, respectively, generated with the aid of an RMI communication at least between the printer 12 and the service notebook 18.

10

A system 56 for maintenance, configuration, diagnosis and operation of a printer 58 is shown in Figure 4. The printer 58 contains an internal control panel 60 that is connected with an input and output controller 62 of the printer 58. Furthermore, a service notebook 64 of a service technician 66 is connected with the input and

15 output controller 62 of the printer 58 via a local area network connection (LAN) 68.

The system 56 according to Figure 4 is shown in Figure 5, whereby a LAN modem 70 is connected with the input and output controller 62 via a LAN network

20 connection 72. What is known as a point-to-point connection is established with a second LAN modem 76 via a telephone network 74 with the aid of the LAN modem 70. The telephone connection 74 is alternately an analog telephone connection or an ISDN telephone connection. A service notebook 64 of the service technician 66 is connected with the LAN modem 76 via a network

25 connection 78. The LAN modem 76, the service notebook 64 and the service technician 66 are located in a service center 80 of the printer manufacturer. With the aid of the service notebook 64, the service technician 66 can implement the same operating, configuration, diagnosis and setting operations as he can implement with a direct network connection 68 between the printer 58 and the

30 service notebook 64. However, given a connection via the telephone network 74

the data transfer rate is significantly less than given a direct network connection 68 between the service notebook 64 and the printer 58.

A block diagram for administering and locating data contained in jar archives is
5   shown in Figure 6. In connection with the Figures 2 through 5, it has already been explained how Java applets and further program data contained in jar archives are executed with the aid of a Java runtime program environment of the browser program module 32. In the exemplary embodiment according to Figure 2, the Java applets are transferred from the storage region 40 to the service notebook 18 with
10   the aid of jar archives. In the arrangement shown in Figure 6, jar archives can be stored in an archive cache region 82 of the service notebook 64. The arrangement according to Figure 6 contains a browser program module 84 for output of a graphical user interface 86. At least parts of the graphical user interface 96 are generated via execution of a Java applet 88.
15

The browser program module 84 executes a hypertext of an HTML page in which is contained a program call of a Java applet 88. After the call of the Java applet 88, it is executed with the aid of the Java runtime program environment 90 (which is also designated as a Java Runtime Environment). What are known as classes made
20   up of at least one jar archive are loaded for execution of the Java applet 88. For the case that the Classloader does not find the class in a default archive, the request is transferred for the downloading of a hierarchically subordinate Classloader that then searches for the requested class in a further preset index. The Java runtime program environment contains a BootstrapClassloader 92 for loading of classes
25   that are contained in a basic configuration of the Java runtime program environment 90. An ExtensionClassloader 94 serves for loading of classes that expand the functional scope of the basic classes and whose access rights is [sic] not limited by a right concept contained in the Java runtime program environment 90. An ApplicationClassloader 96 is also provided that serves for loading of applet- or,
30   respectively, application-specific classes.

The classes that are loaded with the ApplicationClassloader 96 have been specially created for the application or, respectively, for the applet and are not contained in basic version of the Java runtime program environment 90. The ApplicationClassloader 96 serves for loading the archive data and source data of

5   all sources and archives contained in the Java applet 88. Furthermore, a CustumURLClassloader [sic] 104 is provided that is inventively provided in addition to the BootstrapClassloader 92, the ExtensionClassloader 94 and the ApplicationClassloader 96. The CustumURLClassloader [sic] 104 serves for the loading of applet- or, respectively, application-specific classes and program data

10   that are stored in storage region 82 that can be preset. With the help of the CustumURLClassloader [sic] 104, it is possible to read out these classes from any storage region that can be preset, for example from indexes 82, without the limitations in the administration of the [sic] and in the access to the classes that exist in other Classloaders.

15

Upon access to a class administered by the CustumURLClassloader [sic] 104, a request of this class is first supplied to what is known as a BootstrapClassloader 92 that searches the preset archive for the class. If the sought class is not contained in the present archive, a further archive is subsequently searched through for the class

20   with an ExtensionClassloader 94. If the class is also not contained in this archive, a storage region is subsequently searched for the requested class by an ApplicationClassloader 96. The ApplicationClassloader 96 has access both to a browser cache 98 and a plug-in cache 100.

25   Both the archive cache 82, the browser cache 98 and the plug-in cache 100 are contained in a fixed-disc storage of a data processing unit, whereby the data processing unit serves as an operating unit and as a service and maintenance computer of the printer, with which service and maintenance computer a data processing unit of the printer is connected via a network connection 102. The

30   service and maintenance computer also contains the computer program module 84 with the Java runtime program environment 90. Data, classes, archives and Java

applets that are transferred from a printer connected via the network 102 to the service and maintenance computer can be cached in both the browser cache 98 and in the plug-in cache 100.

5   As already mentioned, the ApplicationClassloader 96 has access to the browser cache 98 and to the plug-in cache 100. However, the data that are transferred from the printer over the network 102 are stored in the browser cache 98 and in the plug-in cache 100 with reference to the network address of the printer from which they have been transferred. Furthermore, the data stored in the browser cache 98 and in

10   the plug-in cache 100 are deleted after a preset time span. Given a storage of the invoked Java applet in the browser cache 98 of in the plug-in cache 100, identical archives that contain the requested class must be transferred again to the service and maintenance computer given a change of the network address of the printer, for example given a connection via the network 102 of the service and

15   maintenance computer with a second printer identical in construction.

Both the transfer of an identical archive and the caching of the archive with reference to a second network address in the browser cache 98 or in the plug-in cache 100 is time-consuming and consumes storage space. In particular only a

20   data transfer with low data transfer rate is possible over the network 102. When, for example, a telephone network similar to the telephone network according to Figure 5 is provided between the service and maintenance computer and the printer, the data transfer can last multiple minutes and, given storage-intensive archives, up to multiple hours. If the archive with the requested class is not found

25   upon searching of the caches 98 and 100 by the ApplicationClassloader 96, a CustomURLClassloader 104 is inventively invoked with which the archive cache 82 is associated.

In the archive cache 82, the program data are stored independent of the network

30   address of the printed from which they were transferred. The archive cache 82 can also comprise the storage region of a CD-ROM on which a plurality of archives of

different types of printers and different versions of the same printer types are stored. As an alternative or in addition to this, a plurality of archives are stored on the fixed disc with classes, Java applets and [sic] or Java applications for operation and for diagnosis of various printers. Given invocation of the

5   CustomURLClassloader 104, the archive cache 82 is searched for the archive with the class invoked by the browser program module 84. If the respective archive is stored in the archive cache 82, the archive is read out from this archive cache 82 and transferred to the Java runtime program environment 90 that, together with the browser program module 84, opens the archive and determines the necessary data,

10   i.e. the requested class data. With these class data, the executed Java applet 88 can provide necessary operating and/or diagnosis functions in the graphical user interface 86 for diagnosis, configuration or, respectively, operation of the printer.

However, if the CustomURLClassloader 104 determines that no archives with the

15   class data are contained in the archive cache 82, the archive is requested from the printer connected over the network 102 and transferred from this to the Java runtime program environment 90 of the service and maintenance computer, whereby the archive is stored in the archive cache 82 after the transfer. Given a new connection with a printer in which the same archive is required, the archive is

20   then read out from the archive cache 92 by the CustomURLClassloader 104 and a requested class or a Java applet is executed by the Java runtime program environment 90. A new transfer of the archive from the connected printer is then not necessary, even when the printer has a different network address.

25   The archives advantageously contain program data of Java applets or of parts of a Java applet, class data or other data. These archives contain version information of the archive and are signed in order to prevent a manipulation of the archive data and thereby to prevent a manipulation of the printer and the operating unit.

30   If no data or, respectively, only a small data quantity are transferred between the printer and the service and maintenance computer given an existing network

connection 102, the CustomURLClassloader 104 can request from the printer further archives that are currently not necessary and store in the archive cache 82 those that are not yet contained in the printer and in the archive cache 82. Archives are advantageously requested whose designations are contained in the hypertext of

5   the HTML page 86 or in an already-executed Java applet 88.

A jar archive that contains a primary loader with which further classes and program data that are necessary for generation of the graphical user interface 86 are specifically downloaded is preferably first loaded and executed by the printer

10   or from the archive cache 84 by the browser program module 32. A jar archive with an authentication procedure is subsequently loaded for authentication of the data processing system by the printer. A jar archive with Java applets for communication is subsequently loaded between the printer and the service and maintenance computer. Finally a jar archive with data for generation of the basic

15   functions of the graphical user interface 86 is loaded and executed.

After invocation of individual diagnosis and operating functions via the graphical user interface 86, jar archives necessary for provision of the diagnosis or, respectively, of the operating function are then downloaded from the archive cache

20   82 or from the printer. The hypertext of the HTML page or a jar archive that has been transferred from the printer to the service and maintenance computer preferably contains the designations of all jar archives necessary for possible operating and diagnosis functions. These data are alternatively contained in the primary loader. A Java applet preferably contains all program steps necessary for

25   provision of the operating or, respectively, diagnosis function in order to prevent dependencies between individual archives. A time-consuming downloading of further archives that are not inherently necessary or that are not yet necessary is thereby prevented.

30   In other exemplary embodiments, information about the design and content of the graphical user interface 86 are stored in the printer, in particular about the design

of a menu tree, the content of lists and output fields. This stored information is used to accelerate the construction of the graphical user interface 86 given a regeneration of a graphical user interface 86, in that for the time being only the stored information are transferred from the printer to the service and maintenance

5   computer or to an operating unit. Thus, for example, given a new service assignment information stored in this printer can be accessed and a graphical user interface 86 can be built with the stored information.

The loading of individual jar archives from the cache 82 and from the printer can

10   be omitted up to a point in time at which the respective function is then concretely required, or possibly even when the corresponding function is not necessary in the respective operation and/or diagnosis of the printer. Given storage of the information, additional version data are preferably stored that enable it to determine the timeliness of the stored information. The operating, configuration

15   and diagnosis functions in particular concern parameter settings of the printer, the adjustment of error limit values, the adjustment of voltages between individual components of an image generation unit of the printer, status settings of the printer, light barrier test routines, motor start routines and valve test routines.

20   A block diagram with a data processing unit 110 and a printer controller 112 is shown in Figure 7, in which are shown elements for data transfer between the printer controller 112 and the data processing unit 110. The data processing unit 110 is, for example, a service notebook or an operating unit of the printer. The printer controller 112 contains a web server 114 similar to the HTTP server 36

25   according to Figure 2. Furthermore, the printer controller 112 contains a firmware 116. Via execution of the firmware 116 by the printer controller 112, at least one part of the components of the printer is controlled and an RMI server is generated for RMI communication between printer controller 112 and data processing system 110. Furthermore, measurement data and setting data of the printer are determined

30   with the aid of the firmware 116. The measurement data and the setting data are contained in a storage region 118 of the printer controller 112 or in storage regions

of control units (not shown) of the printer controller 112. The firmware reads these measurement data from the storage region 118 and/or from the control units and, with the aid of the RMI server, provides these data for invocation by the data processing system 110. With the help of a browser program module 120, the data

5   processing unit 110 generates a graphical user interface via which data (in particular measurement and setting values as well as operating states and error data) can be displayed.

The browser program module 120 contains a Java runtime program environment

10   provided with the aid of a plug-in program module program environment 122. With the aid of the program environment 122, the browser program module 120 executes program elements such as, for example, Java applets. Data, class data as well as the program data to be executed of a plurality of applets are contained in the archives in a storage region 124 of the data processing unit 110. The storage

15   region 124 preferably comprises a browser cache, a plug-in cache and an archive cache similar to the caches according to Fig. 6. The data transfer between the web server 114 and the browser program module 122 preferably occurs with a hypertext transfer protocol. The transfer of measurement data and setting values of the printer to Java applet 126 executed by the browser program 120 occurs with the

20   aid of an RMI communication between the Java applet and the RMI server that is generated by the execution of the firmware 116. The Java applet provides functions for configuration and diagnosis of the printer controller 112 and of the printer in which the printer controller 112 is contained.

25   The storage content of the browser cache 98, of the plug-in cache 100 and of the archive cache 82 according to Figure 6 is respectively shown in Figure 8. A service and maintenance computer 130 is respectively connected with a first printer 132 and a fifth printer 134 over a data connection. Jar archives are respectively transferred from the printers 132, 134 to the service and maintenance computer in

30   order to provide a graphical user interface as well as operating, configuration, diagnosis and maintenance functions as described in connection with Figure 6 and

Figure 7. A file "x.jar" and a file "y1.jar" is [sic] transferred from the printer 1 to the service and maintenance computer 130. The service and maintenance computer 130 is configured such that the transferred files are stored both in the browser cache, in the plug-in cache and in the archive cache. Both the file "x.jar"

5   and the file "y1.jar" is [sic] stored in the browser cache and in the plug-in cache under specification of the network address IP1 of the printer 1. The file "x.jar" and the file "y1.jar" is [sic] stored in the archive cache without specification a network address. With the help of the connection between the printer 134 and the service and maintenance computer 130, the file [sic] "x.jar", "y5.jar" and "z.jar" is [sic]

10   transferred to the service and maintenance computer 130, whereby the files "x.jar", "y5.jar" and "z.jar" are stored in the browser cache and in the plug-in cache under specification of the IP address IP5. The file "x.jar" (which is contained in the printer 134) and the file "x.jar" (which is contained in the printer 152 [sic]) are identical. Only the files "y5.jar" and "z.jar" are stored in the archive cache of the

15   service and maintenance computer 130 without specification of a network address. If only the archive cache is provided as a single cache for caching of the jar archives, the file "x.jar" must only be transferred once, either from the printer 132 or from the printer 134 to the service and maintenance computer 130.

20   A block diagram is shown in Figure 9 in which the service and maintenance computer 130 according to Figure 8 is connected with a second printer 136. Before the connection of the service and maintenance computer 130 with the second printer 136, the service and maintenance computer 130 has already been connected with the first and fifth printers 132, 134, as already explained in

25   connection with Figure 8. With the help of the connection between the service and maintenance computer 130 and the second printer 136, the program modules "x.jar" and "y2.jar" are transferred from the printer 136 to the service and maintenance computer 130 and stored in the browser cache and in the plug-in cache under reference to the IP address of the printer 136. However, if (as already

30   explained in connection with Figure 8) instead of the browser cache and the plug-in cache an archive cache is inventively provided in which the jar archives are

stored, only the jar archive "y2.jar" not contained in the archive cache is transferred from the printer 136 from the printer 2 [sic] to the service and maintenance computer 130.

5 After the connection of the service and maintenance computer 130 with the printers 132, 134 and 136, seven files with jar archives that have been transferred from the printers 132, 134, 136 to the service and maintenance computer 130 are respectively stored in the browser cache and in the plug-in cache. Only five files are cached in the archive cache, since the files "x.jar" that are stored in the printers

10 132, 134, 136 coincide and do not have to be transferred multiple times from the printers 132, 134, 136 to the service and maintenance computer 130.

Depending on presets in the browser program module, the data stored in the browser cache and in the plug-in cache are deleted after a predetermined span of

15 time. The data and files contained in the archive cache are at least not automatically deleted. If only the archive cache of the service and maintenance computer 130 is used to store the jar archives, the data quantity to be transferred from the printers 132, 134, 136 to the service and maintenance computer 130 can thus be significantly reduced, whereby wait times are avoided. Via a testing of the

20 necessary versions of the jar archives and of the versions of the jar archives stored in the archive cache, it is ensured that the necessary jar archives are loaded and subsequently processed by the browser program module of the service and maintenance computer 130.

25 A block diagram with jar archives for generation of a graphical user interface for configuration and diagnosis of an electrophotographic printer is shown in Figure 10. As already mentioned, the jar archives contain Java applets and/or Java applications as well as further program data necessary for execution of applets and applications. A first jar archive 140 contains at least one Java applet via which a

30 primary loader function is provided. A second jar archive 142 contains at least class data or program data for administration of data of the printer in the browser

program module. A third jar archive 144 contains at least class or program data for provision of communication functions, whereby what is known as a communication layer is generated. The functions needed by the communication layer are provided by the class or program data contained in the jar archive 142.

5

Furthermore, a jar archive 146 is provided that contains at least class or program data for generation of general menu components and the graphic structuring of the graphical user interface. These general menu components access functions of the communication layer. Furthermore, jar archives 148, 150, 152 and 154 are

10   provided that respectively (with the aid of class and program data) provide a menu group and functions of menu points that can be individually invoked. The menu groups 148 through 154 access functions of the general menu components 146. Furthermore, the menu group 152 directly accesses the communication layer 144. The menu groups 148 through 154 have no dependency upon one another.

15

After the connection of the service and maintenance computer with the printer, the jar archive 140 that contains a relatively small data quantity is transferred. As already mentioned, the jar archive 140 contains the primary loader and furthermore contains printer-specific data. The further jar archives 142 through 154 necessary

20   for generation of the graphical user interface are preferably loaded from the each cache 82. The archive cache 82 is located on a fixed disc storage of the service and maintenance computer. These data can thereby be transferred very quickly to the browser program module.

25   The jar archive 142 "ManagerFrame" for administration of data and the jar archive 144 for communication as well as for authentication is [sic] subsequently loaded. Further jar archives 146 through 154 are only loaded after a successful authentication. In the generation of the general menu components with the help of the jar archive 146, information are used that are stored in the printer and are based

30   on settings of an already-configured user interface. The jar archives 148 through 154 are only transferred to the browser program module after a function contained

in the respective menu group has been requested via the graphical user interface. Further jar archives can also be transferred from the printer to the service and maintenance computer that are currently not needed by the browser program module for generation of the graphical user interface. This transfer of the data

5   preferably occurs at points in time in which no further data are to be transferred from the printer to the service and maintenance computer. All jar archives transferred from the printer to the service and maintenance computer are preferably stored in the archive cache 82 when the archives are not yet contained in the archive cache 82 or are only contained in the archive cache 82 in a different

10   program version.

Data can also be transferred from the printer to the service and maintenance computer that contain Java applications, whereby the execution of these Java applications occurs with the help of what is known as a Java Web start technology.

15   In the Java Web start technology, an execution of the Java applet or a Java application occurs not with the help of a browser program module, but rather via execution of a Java network launching protocol file. The applet or, respectively, application loaded by the Web server is thereby executed as a local application on the data processing system of the operating unit or, respectively, of the service

20   notebook. In particular wired and wireless data connections such as Bluetooth, UMTS and DSL are used as data lines for transfer of data.

Although preferred exemplary embodiments are shown and described in detail in the drawings and in the preceding specification, this should be viewed as purely

25   exemplary and not as limiting the invention. It is noted that only the preferred exemplary embodiments are shown and described, and all variations and modifications that presently and in the future lie within the scope of protection of the invention should be protected.

30

Reference list

	10	system
	12	printer
5	14	device controller
	16	input and output controller
	18	service and maintenance computer
	20, 22	network connection
	24	network
10	26	external operating unit
	28	LAN modem
	30	telephone network
	32	display program
	34	Java runtime program environment
15	36	HTTP server
	38, 40	storage region
	42, 44, 46, 48	program functions
	50	RMI server
	52, 54	SNMP agent
20	56	system
	58	printer
	60	operating unit
	62	input and output control unit
	64	service and maintenance computer
25	66	service technician
	68, 72, 78	network connection
	70, 76	LAN modem
	74	telephone network
	80	service control center
30	82, 98, 100	cache
	84	browser program module

	86	graphical user interface
	88	Java applet
	90	Java runtime program environment
	92	BootstrapClassloader
5	94	ExtensionsClassloader
	96	ApplicationClassloader
	102	network for the printer
	104	CustomURLClassloader
	110	data processing unit
10	112	printer controller
	114	Web server
	116	firmware
	118	data storage
	120	browser program module
15	122	Java runtime program environment
	124	data storage
	126	Java applet
	130	service and maintenance computer
	132, 134, 136	printer
20	140 through 154	jar archives